

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:
RAJAGOPALAN et al.

Serial No.: 10/731,063

Confirmation No.: 3734

Filed: December 9, 2003

For: UPLOADING TCP FRAME
DATA TO USER BUFFERS
AND BUFFERS IN SYSTEM
MEMORY

§
§
§
§
§
§
§
§
§
§

Group Art Unit: 2182

Examiner: Jasjit S. Vidwan

MAIL STOP APPEAL BRIEF-PATENTS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Dear Sir:

The Appellant submits this Appeal Brief to the Board of Patent Appeals and Interferences on appeal from the decision of the Examiner of Group Art Unit 3734 dated September 4, 2008, finally rejecting claims 1-6, 8-10, 21, 22, and 24-30 (the "Final Office Action"). The final rejection of claims 1-6, 8-10, 21, 22, and 24-30 is appealed. The Commissioner is hereby authorized to charge Deposit Account No. 20-0782/NVDA/P000785/SW for any fees necessary to make this Appeal Brief timely and acceptable to the Office.

TABLE OF CONTENTS

1.	Identification Page.....	1
2.	Table of Contents	2
3.	Real Party in Interest	3
4.	Related Appeals and Interferences	4
5.	Status of Claims	5
6.	Status of Amendments	6
7.	Summary of Claimed Subject Matter	7
8.	Grounds of Rejection to be Reviewed on Appeal	10
9.	Arguments	11
10.	Conclusion	14
11.	Claims Appendix	15
12.	Evidence Appendix	18
13.	Related Proceedings Appendix	19

Real Party in Interest

The present real party in interest is NVIDIA Corporation, a corporation of the State of Delaware, having a place of business at 2701 San Tomas Expressway, Santa Clara, CA 95050.

Related Appeals and Interferences

Applicant asserts that no other appeals or interferences are known to the Applicant, the Applicant's legal representative, or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

Status of Claims

Claims 1-6, 8-10, 21, 22, and 24-30 are pending in the application. Claims 1-20 were originally presented in the application. Claims 7, 11-20, and 23 are cancelled. Claims 1-6, 8-10, 21, 22, and 24-30 stand finally rejected as discussed below. The final rejections of claims 1-6, 8-10, 21, 22, and 24-30 are appealed. The pending claims are shown in the attached Claims Appendix.

Status of Amendments

All claim amendments prior to and including the Final Office Action have been entered by the Examiner.

Summary of Claimed Subject Matter

Claimed embodiments include an apparatus for uploading frame data to system memory (*see, e.g.*, claims 1-6, 8-9, 21-22, and 28-29) and a method for uploading frame data including Transmission Control Protocol (TCP) payload data to system memory (*see, e.g.*, claims 10, 24-27, and 30).

Claim 1 recites an apparatus for uploading frame data to system memory. As claimed, the system includes (i) a CPU coupled to the system memory and configured to execute an application program, the CPU executing a Transmission Control Protocol (TCP) stack which includes code to complete at least some TCP processing, (ii) a hardware subsystem preconfigured to process all frames related to one or more connections delegated by the TCP stack to produce frame data and to upload the frame data to a user buffer in the system memory allocated to the application program, if the user buffer is available, and to a legacy buffer in a portion of the system memory that is not allocated to the application program, if the user buffer is not available, where frame data uploaded to the legacy buffer is copied from the portion of the system memory that is not allocated to the application program to the system memory that is allocated to the application program, and (iii) the system memory including a connection table (CT) storing data for all active connections with system including delegated connections, reads and writes to the CT being made directly through the hardware, the hardware subsystem being further preconfigured to request legacy processing by the TCP stack of the frames of the delegated connections.

The limitations of a CPU coupled to the system memory and configured to execute an application program and the CPU executing a TCP stack which includes code to complete at least some TCP processing is supported in *Application*, paragraph [0039], and elements 110, 120, 220, and 130 of Figures 2A and 2B. The limitations of a hardware subsystem preconfigured to process all frames related to one or more connections delegated by the TCP stack to produce frame data and to upload the frame data to a user buffer in the system memory allocated to the application program, if the user buffer is available, and to a legacy buffer in a portion of the system memory that is not allocated to the application program, if the user buffer is not available, where frame data uploaded to the legacy buffer is copied from the portion of the system memory that

is not allocated to the application program to the system memory that is allocated to the application program is supported in *Application*, paragraph [0054], the last sentence of paragraph [0050], elements 427, 430, and 428 of Figure 4B, and Figures 5A and 5B. The limitations of the system memory including a CT storing data for all active connections with system including delegated connections, reads and writes to the CT being made directly through the hardware, the hardware subsystem being further preconfigured to request legacy processing by the TCP stack of the frames of the delegated connections, is supported in *Application*, paragraph [0044] and element 245 of Figures 2A and 2B.

Claim 10 recites a method for uploading frame data including Transmission Control Protocol (TCP) payload data to system memory. As claimed, the method includes (i) processing a frame to produce frame data, (ii) uploading the frame data to either a portion of system memory comprising a user buffer allocated to an application program, if the user buffer is available, and to a legacy buffer in a portion of the system memory that is not allocated to the application program for separate TCP processing by a TCP stack executing on a CPU, if the user buffer is not available, (iii) copying frame data uploaded to the legacy buffer from the portion of the system memory that is not allocated to the application program to the system memory that is allocated to the application program, and (iv) utilizing hardware separate from the CPU which does the TCP processing to partially process the frame and determine whether the frame was delegated by the separate TCP processing.

The limitations of processing a frame to produce frame data is supported in *Application*, paragraphs [0053] - [0054], and elements 424 and 426 of Figure 4B. The limitations of uploading the frame data to either a portion of system memory comprising a user buffer allocated to an application program, if the user buffer is available, and to a legacy buffer in a portion of the system memory that is not allocated to the application program for separate TCP processing by a TCP stack executing on a CPU, if the user buffer is not available is supported in *Application*, paragraph [0054], elements 427, 430, and 428 of Figure 4B, and Figures 5A and 5B. The limitations of copying frame data uploaded to the legacy buffer from the portion of the system memory that is not allocated to the application program to the system memory that is allocated to the

application program, is supported in *Application*, the last sentence of paragraph [0050]. The limitation of utilizing hardware separate from the CPU which does the TCP processing to partially process the frame and determine whether the frame was delegated by the separate TCP processing is supported in *Application*, paragraphs [0032] and [0044].

An important aspect of embodiments of Appellants' invention is that frame data is uploaded to a portion of system memory comprising a user buffer allocated to an application program, if the user buffer is available, and the frame data is uploaded to a legacy buffer in a portion of the system memory that is not allocated to the application program for separate TCP processing by a TCP stack executing on a CPU, if the user buffer is not available.

Grounds of Rejection to be Reviewed

1. Claims 1-3, 10, and 29-30 stand rejected under 35 U.S.C. §103(a) as unpatentable over Boucher (6,334,153) in view of Elzur (6,629,125).
2. Claims 4-6, 8-9, 21-22, and 24-28 stand rejected under 35 U.S.C. §103(a) as unpatentable over Boucher (6,334,153) and Elzur (6,629,125) in further view of Adams (6,775,693).

ARGUMENT

Obviousness of Claims 1-3, 10, and 29-30 by Boucher and Elzur

Claim 1 recites the limitations of a hardware subsystem preconfigured to upload the frame data to a user buffer in the system memory allocated to the application program, if the user buffer is available, and to a legacy buffer in a portion of the system memory that is not allocated to the application program, if the user buffer is not available. The combination of Boucher and Elzur fails to teach or suggest these limitations.

As is clearly recited in claim 1 and set forth in step 427 of Figure 4B as well as in paragraph [0054] of the present application, when a user buffer is not available the frame data is uploaded to a legacy buffer. The user buffers are allocated in an application memory space and legacy buffers are allocated in a driver memory space, as shown in Figures 2A, 2B, 5A, and 5C of the present application. Importantly, the application memory space is separate from the driver memory space. Those skilled in the art would recognize that separating the portion of system memory that is allocated to an application program and the portion of system memory that is allocated to a driver, as claimed, prevents corruption of the data read and/or written by the application program and driver.

In contrast to the claimed approach, Boucher teaches a mechanism of queuing packets not eligible for fast-path processing until a user buffer is available. In particular, Boucher does not disclose redirecting the data to a legacy buffer when the user buffer is unavailable (or full). Thus, the Examiner relies on Elzur for teaching redirecting data to a legacy buffer when the user buffer is not available. However, a review of Elzur shows that this reference also does not teach or suggest these missing limitations.

More specifically, Elzur discloses a system for uploading frame data to user buffers that are allocated to an application program. The Examiner equates the claimed user buffer with memory buffer 304 disclosed in Elzur. Elzur teaches that memory buffer 304 is associated with the application (see Figure 7 and column 5, lines 2-5 of Elzur). The Examiner equates the claimed legacy buffer with element 308 disclosed in Elzur. However, such an interpretation is inconsistent with the specification of Elzur. Elzur teaches that element 308 resides within memory buffer 304 (see column 9, lines

44-45). Thus, like memory buffer 304, element 308 is also within memory that is allocated to the application program. Importantly, Elzur teaches that region 308 is reserved within buffer 304 to store a packet that is received out of sequence and that region 308 is sized to store the out of sequence packet (see col. 9, lines 52-55 of Elzur). Nowhere does Elzur teach or suggest that any packets are uploaded to any type of a legacy buffer when a user buffer (i.e., buffer 304) is available, as expressly claimed.

In sum, element 308 is not functionally equivalent to the claimed legacy buffer within a portion of the system memory that is allocated to a driver. The distinction that a user buffer in the system memory allocated to the application program and a legacy buffer in a portion of the system memory that is not allocated to the application program is clearly articulated in claim 1. Claim 1 was previously amended to clarify this distinction between a user buffer and a legacy buffer with the intent of overcoming the Elzur and Boucher references, which Applicants believe they have successfully done.

As the foregoing illustrates, the combination of Boucher and Elzur fails to teach or suggest the limitation of uploading the frame data to a legacy buffer in a portion of the system memory that is not allocated to the application program, if the user buffer is not available, as explicitly recited in claim 1. Similarly, the combination of Boucher and Elzur fails to teach or suggest the limitation of uploading the frame data to a legacy buffer in a portion of the system memory that is not allocated to the application program for separate TCP processing by a TCP stack executing on a CPU, if the user buffer is not available, as explicitly recited in claim 10. Therefore, no combination of the Boucher and Elzur can render either claim 1 or claim 10 obvious. For these reasons, Applicant submits that claims 1 and 10 are in condition for allowance and respectfully request withdrawal of the 35 U.S.C. § 103(a) rejection of these claims. Since claims 2-6, 8-9, 21, 22, 28, and 29 depend from allowable claim 1 and claims 24-27 and 30 depend from allowable claim 10, these claims also are in condition for allowance.

Obviousness of Claims 4-6, 8-9, 21-22, and 24-28 over Boucher, Elzur, and Adams

Adams fails to cure the deficiencies of Boucher and Elzur set forth above.

The Adams reference teaches transferring data between application buffers over a network. Adams is silent regarding two different types of buffers in system memory

that are configured to receive data. Adams also fails to teach or suggest the limitations of uploading frame data to a user buffer in the system memory allocated to the application program, if the user buffer is available, and to a legacy buffer in a portion of the system memory that is not allocated to the application program if the user buffer is not available. Thus, Adams fails to cure the deficiencies of Boucher and Elzur relative to claims 1 and 10. For this reason, the combination of Boucher, Elzur, and Adams cannot render any of claims 1-6, 8-10, 21, 22, 24-27, 28, 29 and 30 obvious either.

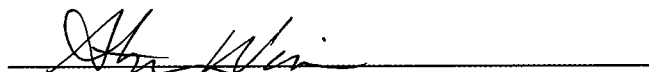
For the aforementioned reasons, all of the claims currently pending in the application are therefore patentable over Boucher, Elzur, and Adams. In view of these clear distinctions, reconsideration and allowance of all the claims is respectfully requested.

CONCLUSION

The Examiner errs in finding that:

- Claims 1-3, 10, and 29-30 are unpatentable over Boucher and Elzur.
- Claims 4-6, 8-9, 21-22, and 24-28 are unpatentable over Boucher and Elzur in view of Adams.

Respectfully submitted,



Stephanie Winner
Registration No. 52,371
PATTERSON & SHERIDAN, L.L.P.
3040 Post Oak Blvd. Suite 1500
Houston, TX 77056
Telephone: (650) 330-2310
Facsimile: (650) 330-2314
Agent for Applicants

CLAIMS APPENDIX

1. (Previously Presented): A system for uploading frame data to system memory, the system comprising:

a CPU coupled to the system memory and configured to execute an application program,

the CPU executing a Transmission Control Protocol (TCP) stack which includes code to complete at least some TCP processing;

a hardware subsystem preconfigured to process all frames related to one or more connections delegated by the TCP stack to produce frame data and to upload the frame data to a user buffer in the system memory allocated to the application program, if the user buffer is available, and to a legacy buffer in a portion of the system memory that is not allocated to the application program, if the user buffer is not available, wherein frame data uploaded to the legacy buffer is copied from the portion of the system memory that is not allocated to the application program to the system memory that is allocated to the application program; and

the system memory including a connection table (CT) storing data for all active connections with system including delegated connections, reads and writes to the CT being made directly through the hardware,

the hardware subsystem being further preconfigured to request legacy processing by the TCP stack of the frames of the delegated connections.

2. (Original): The system of claim 1, wherein the frame data is payload data.

3. (Previously Presented): The system of claim 1, wherein the TCP Stack provides the hardware with a physical address corresponding to a user buffer.

4. (Previously Presented): The system of claim 1, wherein the hardware is configured to process frames to produce partially processed frame data and upload at least a portion of the partially processed frame data to a legacy buffer, the hardware maintaining the CT entry point for each received packet.

5. (Previously Presented): The system of claim 4, wherein the legacy buffer storing the portion of the partially processed frame data is stored in the portion of system memory that is not allocated to the application program.

6. (Previously Presented): The system of claim 2, wherein the hardware is configured to upload at least a portion of the payload data to the legacy buffer.

7. (Cancelled)

8. (Original): The system of claim 6, wherein a software driver provides the hardware with a tag corresponding to a location of the legacy buffer.

9. (Previously Presented): The system of claim 8, wherein the hardware is configured to transmit the tag to the software driver.

10. (Previously Presented): A method of uploading frame data including Transmission Control Protocol (TCP) payload data to system memory, the method comprising:

processing a frame to produce frame data;

uploading the frame data to either a portion of system memory comprising a user buffer allocated to an application program, if the user buffer is available, and to a legacy buffer in a portion of the system memory that is not allocated to the application program for separate TCP processing by a TCP stack executing on a CPU, if the user buffer is not available;

copying frame data uploaded to the legacy buffer from the portion of the system memory that is not allocated to the application program to the system memory that is allocated to the application program; and

utilizing hardware separate from the CPU which does the TCP processing to partially process the frame and determine whether the frame was delegated by the separate TCP processing.

11 – 20. (Cancelled)

21. (Previously Presented): The system of claim 1 wherein the hardware is configured to pause incoming frame data to determine whether a frame is invalid, the invalid frame being stored in the legacy buffer for legacy processing.
22. (Previously Presented): The system of claim 4, wherein the TCP stack is configured to process the frame data uploaded to the legacy buffer by the hardware.
23. (Cancelled)
24. (Previously Presented): A method as claimed in claim 10 wherein the TCP stack completes processing of the partially processed frame stored in the legacy buffer.
25. (Previously Presented): A method as claimed in claim 24 wherein the partial processing of the frame produces partially processed frame and header data.
26. (Previously Presented): A method as in claim 10 wherein the user buffer is defined as not available when the processed frame portion exceeds a start up limit value associated with the delegated connection carrying the frame being processed.
27. (Previously Presented): A method as in claim 10 wherein the uploaded frame data includes TCP payload data.
28. (Previously Presented): A system as in claim 1 wherein the hardware accesses the CT using the distribution port field of a frame as a direct index into the CT.
29. (Previously Presented): The system of claim 1, wherein the portion of system memory that stores the legacy buffer is allocated to a software driver.
30. (Previously Presented): The method of claim 10, wherein the portion of system memory that stores the legacy buffer is allocated to a software driver.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None